PREMIUM AUDIT

NAME OF PROJECT	Ape Finance
TOKEN BEP20 ADDRESS	0x1f00abE7aBce44e4b5319c5b7825EB3A4A237196
WEBSITE / TELEGRAM	http://apefinance.financial

DISCLAIMER: PLEASE READ FULL AUDIT

IMPORTANT DISCLAIMER

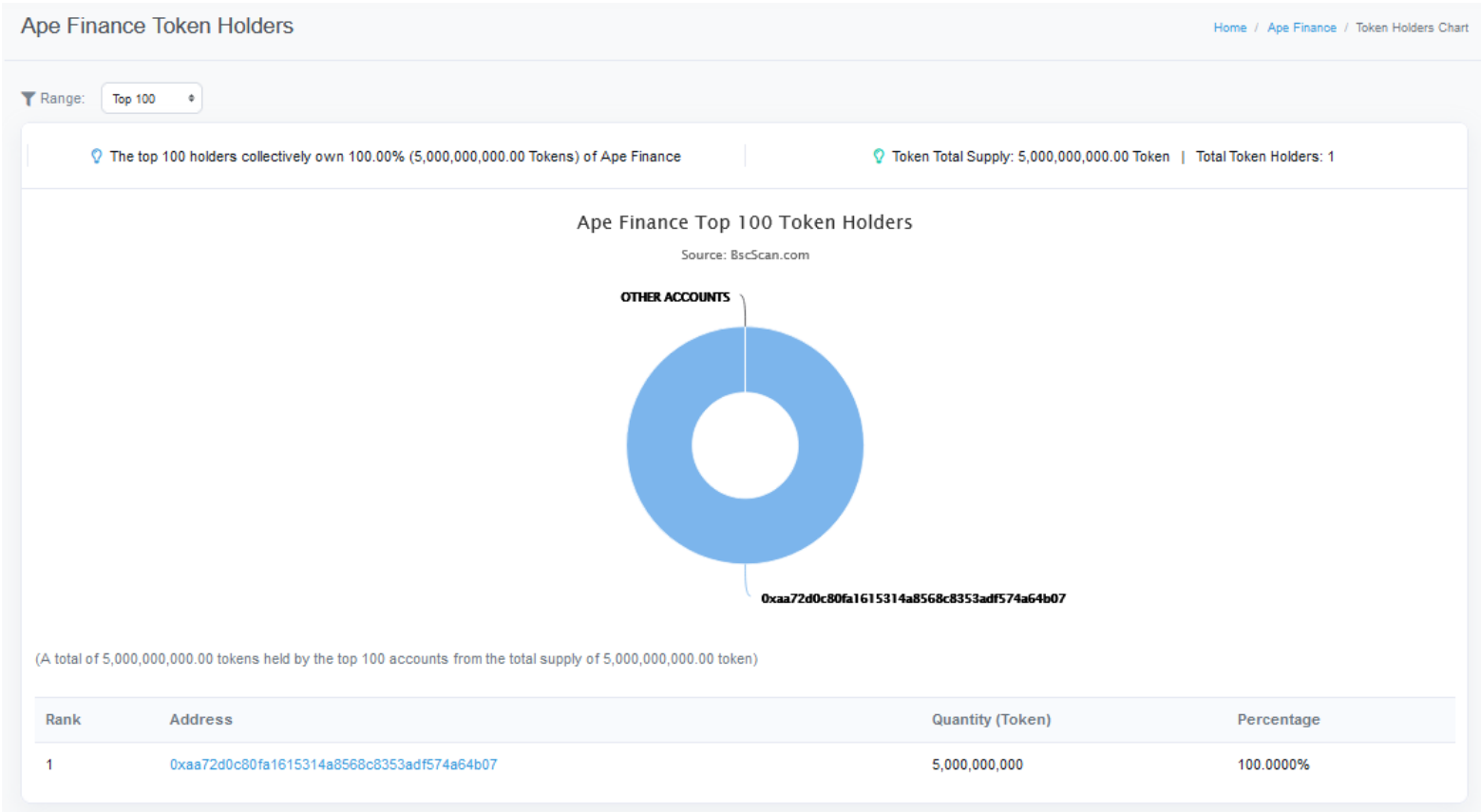|		
I.	**NOT RESPONSABLE**	Our audit report is not responsible for anything if the project audited turn into a scam.
II.	**NOT GUARANTEE**	Analytix Audit in no way guarantees that a project will not remove liquidity, sell off team tokens, or exit scam.
III.	**INFORMATION**	Analytix Audit do a research and provides public information about the project in an easy-to-understand format for the common person.
IV.	**AUDIT AGREE**	Agreeing to an audit in no way guarantees that a team will not remove all liquidity ("Rug Pull"), remove liquidity slowly, sell off tokens, quit the project, or completely exit scam
V.	**ENCOURAGE INVESTMENT**	Analytix Audit in no way takes responsibility for any losses, nor does Analytix Audit encourage any speculative investments
VI.	**INFORMATION**	The information provided in this audit is for information purposes only and should not be considered investment advice. Analytix Audit does not endorse, recommend, support, or suggest any projects that have been audited.

TOKENOMICS - OVERVIEW

Ape Finance Token Holders

Range: Top 100

The top 100 holders collectively own 100.00% (5,000,000,000.00 Tokens) of Ape Finance | Token Total Supply: 5,000,000,000.00 Token | Total Token Holders: 1

Ape Finance Top 100 Token Holders

Source: BscScan.com

OTHER ACCOUNTS

0xaa72d0c80fa1615314a8568c8353adf574a64b07

(A total of 5,000,000,000.00 tokens held by the top 100 accounts from the total supply of 5,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xaa72d0c80fa1615314a8568c8353adf574a64b07	5,000,000,000	100.0000%

• More than 15% of the tokens are unlocked

(This comment refers to when the audit was carried out maybe team locked tokens later)

CONTRACT CODE - OVERVIEW


```
520    * Requirements:
521    *
522    * - the calling contract must have an ETH balance of at least `value`.
523    * - the called Solidity function must be `payable`.
524    *
525    * _Available since v3.1._
526    */
527    function functionCallWithValue(
528        address target,
529        bytes memory data,
530        uint256 value
531    ) internal returns (bytes memory) {
532        return functionCallWithValue(target, data, value, "Address: low-level call with value failed");
533    }
534
535    /**
536    * @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256-}[`functionCallWithValue`], but
537    * with `errorMessage` as a fallback revert reason when `target` reverts.
538    *
539    * _Available since v3.1._
540    */
541    function functionCallWithValue(
542        address target,
543        bytes memory data,
544        uint256 value,
545        string memory errorMessage
546    ) internal returns (bytes memory) {
547        require(address(this).balance >= value, "Address: insufficient balance for call");
548        require(isContract(target), "Address: call to non-contract");
549
550        (bool success, bytes memory returndata) = target.call{value: value}(data);
```

QUICK REVIEW

• Owner can change swapback

• Owner can change rebase settings

• Owner can toggle trading status

• Owner can't set Tx Limits.

• Owner can't modify fees

Owner Privileges:

```
520      * Requirements:
521      *
522      * - the calling contract must have an ETH balance of at least `value`.
523      * - the called Solidity function must be `payable`.
524      *
525      * _Available since v3.1._
526      */
527     function functionCallWithValue(
528         address target,
529         bytes memory data,
530         uint256 value
531     ) internal returns (bytes memory) {
532         return functionCallWithValue(target, data, value, "Address: low-level call with value failed");
533     }
534
535     /**
536      * @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256-}[`functionCallWithValue`], but
537      * with `errorMessage` as a fallback revert reason when `target` reverts.
538      *
539      * _Available since v3.1._
540      */
541     function functionCallWithValue(
542         address target,
543         bytes memory data,
544         uint256 value,
545         string memory errorMessage
546     ) internal returns (bytes memory) {
547         require(address(this).balance >= value, "Address: insufficient balance for call");
548         require(isContract(target), "Address: call to non-contract");
549
550         (bool success, bytes memory returndata) = target.call{value: value}(data);
```

function renounceOwnership() public virtual onlyOwner {
function transferOwnership(address newOwner) public virtual onlyOwner {
function manualRebase() external onlyOwner {
function setAutomatedMarketMakerPair(address _pair, bool _value)
function setInitialDistributionFinished(bool _value) external onlyOwner {
function setFeeExempt(address _addr, bool _value) external onlyOwner {
function setTargetLiquidity(uint256 target, uint256 accuracy) external onlyOwner {
function setSwapBackSettings(
function setFeeReceivers(address _liquidityReceiver, address _treasuryReceiver, address _riskFreeValueReceiver)
external onlyOwner {
function clearStuckBalance(address _receiver) external onlyOwner {
function rescueToken(address tokenAddress, uint256 tokens) external onlyOwner returns (bool success) {
function setAutoRebase(bool _autoRebase) external onlyOwner {
function setRebaseFrequency(uint256 _rebaseFrequency) external onlyOwner {
function setRewardYield(uint256 _rewardYield, uint256 _rewardYieldDenominator) external onlyOwner {
function setFeesOnNormalTransfers(bool _enabled) external onlyOwner {
function setIsLiquidityInBnb(bool _value) external onlyOwner {
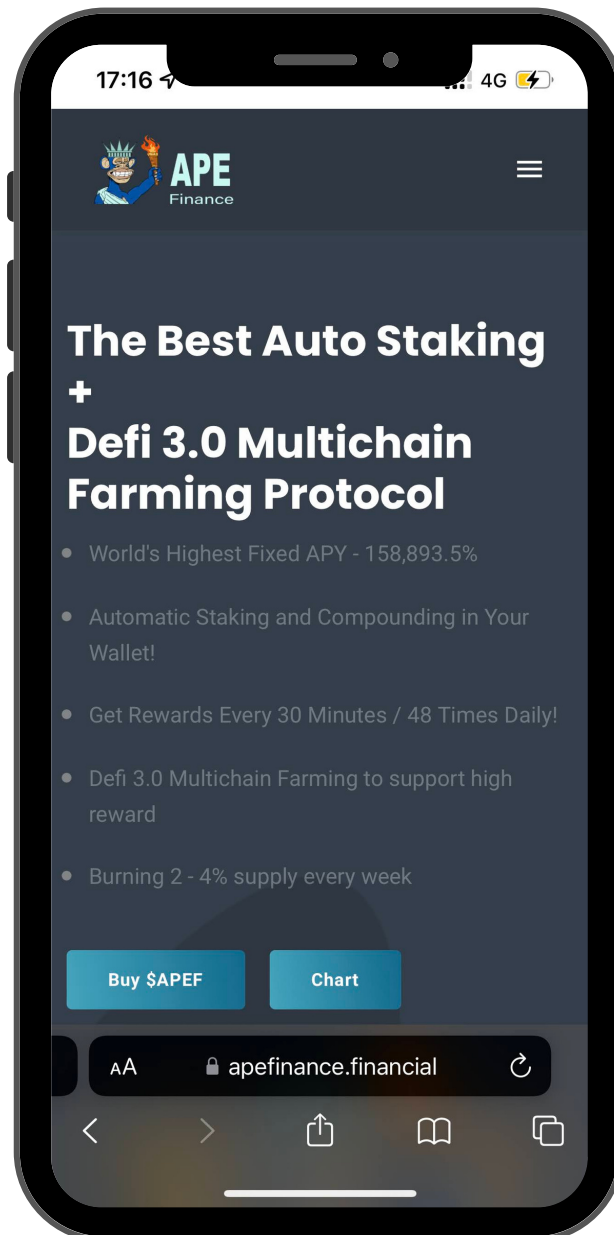function setNextRebase(uint256 _nextRebase) external onlyOwner {

Issue Checking:

N°	Issue description.	Checking Status.
1	Compiler Errors	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module	Passed
21	Fallback function security.	Passed

Audit Result:

PASSED

WEBSITE - OVERVIEW

⏱ Domain Registration: 2022-03-14

5 days old

SSL CERTIFICATE: A

(https://www.ssllabs.com)

SOCIAL MEDIA - OVERVIEW

APEFINANCEBSC

APEFINANCEBSC

APEFINANCE.FINANCIAL